

## CLAIMS

I claim:

1. A method for avoiding base address collisions when loading process components into memory comprising:

detecting that a process has been loaded from a persistent storage into a memory of a computer system, wherein said process is comprised of one or more original components;

for each of said original components, determining whether the original component is loaded into the memory at its preferred base address or at an alternate base address;

in response to determining that a selected original component has been loaded into the memory at the alternate address, creating an alternate component comprising a duplicate of the selected original component with its preferred base address reset to the alternate base address; and

in response to a subsequent request to load the selected original component into memory, loading the alternate component into memory.

2. A computer readable medium having stored thereon computer executable instruction for performing the method of claim 1.

3. The method of claim 1, wherein detecting that the process has been loaded from the persistent storage into the memory further comprises the steps of:

determining whether the process has been loaded into the memory within a specified time limit; and

if the process has not been loaded into the memory within the specified time limit, awaiting a detection that another process has been loaded from the persistent storage into the memory.

4. The method of claim 1, wherein determining whether the original component is loaded into the memory at its preferred base address or at the alternate base address comprises:

comparing the original component's in-memory base address to the preferred base address specified in an on-disk representation of the original component; and

determining whether the in-memory base address is the same as the preferred base address specified in the on-disk representation of the original component

5. The method of claim 4, further comprising the steps of:

in response to determining that the in-memory base address is not the same as the preferred base address specified in the on-disk representation of the original component, recording relocation information for the original component to a relocation file;

wherein said relocation information identifies at least the original component, the preferred base address of the original component and the alternate base address of the original component; and

wherein the relocation information is retrieved from the relocation file in order to create the alternate component.

6. The method of claim 1, wherein the step of loading the alternate component into memory in response to a subsequent request to load the selected original component into memory comprises:

intercepting said request;

modifying said request by substituting therein an identifier of the alternate component for an identifier of the original component; and

passing the modified request to a file system driver for processing.

7. A computer readable medium having stored thereon computer executable instruction for performing the method of claim 6.

8. The method of claim 1, wherein the alternate component comprises an alternate data stream of the original component.

9. The method of claim 1, further comprising the steps of:

creating a second alternate component comprising a duplicate of the selected original component with its preferred base address reset to the alternate base address; and

binding the second alternate component to the alternate base address.

10. The method of claim 9, wherein the step of loading the alternate component into memory in response to the subsequent request to load the selected original component comprises loading either the alternate component or the second alternate component into memory.

11. The method of claim 10, wherein a look-up table is consulted to determine which of the alternate component or the second alternate component to load into memory.

12. The method of claim 9, wherein the alternate component and the second alternate component each comprises an alternate data stream of the selected original component.

13. A computer readable medium having stored thereon computer executable instruction for performing the method of claim 9.

14. A method for avoiding base address collisions when loading process components into memory comprising:

monitoring the loading of a plurality of original components from a persistent storage into a memory of a computer system;

detecting that one or more relocated original component has been loaded into memory at an alternate base address instead of its preferred base address;

creating at least a first alternate component corresponding to each relocated original component, wherein each first alternate component comprises a copy of its corresponding relocated original component and wherein the preferred base address of each first alternate component is reset to be the alternate base address of its corresponding relocated original component; and

in response to a request to load a selected original component into memory, determining whether the selected original component has at least one corresponding alternate component and if so, loading one of the corresponding alternate components into memory instead of the selected original component.

15. A computer readable medium having stored thereon computer executable instruction for performing the method of claim 14.

16. The method of claim 14, further comprising the step of creating a second alternate component corresponding to each relocated original component comprising another duplicate of the selected original component with its preferred base address reset to the alternate base address and wherein the second alternate component is bound to the alternate base address; and

wherein loading one of the corresponding alternate components into memory instead of the selected original component comprises loading either the first alternate component or the second alternate component into memory.

17. The method of claim 16, wherein the first alternate component and the second alternate component each comprises an alternate data stream of the corresponding relocated original component.

18. The method of claim 16, wherein the step of loading one of the corresponding alternate components into memory in response to the request to load the selected original component comprises:

- intercepting said request;
- modifying said request by substituting therein an identifier of the first alternate component or an identifier of the second alternate component for an identifier of the selected original component; and
- passing the modified request to a file system driver for processing.

19. A computer readable medium having stored thereon computer executable instruction for performing the method of claim 20.

20. The method of claim 14, wherein the step of loading one of the corresponding alternate components into memory in response to the request to load the selected original component comprises:

- intercepting said request;
- modifying said request by substituting therein an identifier of the first alternate component for an identifier of the selected original component; and
- passing the modified request to a file system driver for processing.

21. A computer readable medium having stored thereon computer executable instruction for performing the method of claim 20.

22. The method of claim 14, wherein the first alternate component comprises an alternate data stream of the relocated original component.

23. A system for dynamically setting an optimal base address for a component of a process comprising:

a persistent storage for storing a process, the process comprising one or more original components;

a memory being logically divided into a plurality of in-memory addresses; and

a processor for executing computer-executable instructions for:

detecting that one or more relocated original components has been loaded into memory at an alternate base address instead of its preferred base address;

creating an alternate component corresponding to each relocated original component, wherein the alternate component comprises a copy of its corresponding relocated original component and wherein the preferred base address of each alternate component is reset to be the alternate base address of its corresponding relocated original component; and

in response to a request to load a selected original component into memory, determining whether the selected original component has a corresponding alternate component and if so, loading the corresponding alternate component into memory instead of the selected original component.

24. The system of claim 23, wherein the computer-executable instructions for responding to the request to load a selected original component into memory comprise a file system filter driver.

25. The system of claim 23, wherein the alternate component comprises an alternate data stream of the relocated original component.

26. The system of claim 23, wherein the processor executes further computer-executable instructions for recording relocation information to a file;

wherein said relocation information identifies at least the relocated original component, the alternate base address of the relocated original component and the preferred base address of the relocated original component; and

wherein the relocation information is retrieved from the relocation file in order to create the alternate component.